Date=24/12/2020
Lecture By= Akash Handa
Subject ⇒ M-tube Project Roadmap

| IN PREVIOUS LECTURE (QUICK RECAP) Date-23/12/2020 | In Today's Lecture (Overview) |
|---|---|
| Introducing Hooks<br>    When to use a Hooks<br>    Rules of Hooks<br>        1. Only call Hooks at the top level<br>        2. Only call Hooks from React functions<br>    Prerequisites for React Hooks<br>    Hooks State<br>    Hooks Effect<br>        Effects without Cleanup<br>        Effects with Cleanup<br>    Custom Hooks | M-tube Project<br><br>React-google-login<br>    Stay Logged in<br>    onSuccess callback<br><br>YouTube Data API Overview<br>    Introduction<br>        Before you start<br>    Resources and resource types |

# M-tube Project

Next week We are gonna Start Project called m-tube
Today instructor discussed The roadmap About The Project
So for Complete Reference You Have To watch The today's Lecture
Here's The Lecture Link
http://d3w35yba4tosuh.cloudfront.net/06be4580-45a7-11eb-b7f2-35f055f313bb.mp4

# React-google-login

A Google oAUth Sign-in / Log-in Component for React

## Install

```
npm install react-google-login
```

How to use

```
ReactDOM.render(

  <GoogleLogin

clientId="658977310896-knrl3gka66fldh83dao2rhgbblmd4un9.apps.googleusercontent.com"

    buttonText="Login"

    onSuccess={responseGoogle}

    onFailure={responseGoogle}

    cookiePolicy={'single_host_origin'}

  />,

  document.getElementById('googleButton')

);
```

## Stay Logged in

`isSignedIn={true}` attribute will call `onSuccess` callback on load to keep the user signed in.

```
<GoogleLogin
```

```
clientId="658977310896-knrl3gka66fldh83dao2rhgbblmd4un9.apps.googleusercon
tent.com"

  onSuccess={responseGoogle}

  isSignedIn={true}

/>
```

## onSuccess callback

If responseType is not 'code', callback will return the GoogleAuth object.

If responseType is 'code', callback will return the authorization code that can be used to retrieve a refresh token from the server.

If you use the hostedDomain param, make sure to validate the id_token (a JSON web token) returned by Google on your backend server:

1. In the `responseGoogle(response) {...}` callback function, you should get back a standard JWT located at `response.tokenId` or `res.getAuthResponse().id_token`
2. Send this token to your server (preferably as an `Authorization` header)
3. Have your server decode the id_token by using a common JWT library such as jwt-simple or by sending a GET request to `https://www.googleapis.com/oauth2/v3/tokeninfo?id_token=YOUR_TOKEN_HERE`
4. The returned decoded token should have an `hd` key equal to the hosted domain you'd like to restrict to.

# YouTube Data API Overview

## Introduction

This document is intended for developers who want to write applications that interact with YouTube. It explains basic concepts of YouTube and of the API itself. It also provides an overview of the different functions that the API supports.

**Before you start**

1. You need a Google Account to access the Google API Console, request an API key, and register your application.
2. Create a project in the Google Developers Console and obtain authorization credentials so your application can submit API requests.
3. After creating your project, make sure the YouTube Data API is one of the services that your application is registered to use:
   1. Go to the API Console and select the project that you just registered.
   2. Visit the Enabled APIs page. In the list of APIs, make sure the status is **ON** for the **YouTube Data API v3**.
4. If your application will use any API methods that require user authorization, read the authentication guide to learn how to implement OAuth 2.0 authorization.
5. Select a client library to simplify your API implementation.
6. Familiarize yourself with the core concepts of the JSON (JavaScript Object Notation) data format. JSON is a common, language-independent data format that provides a simple text representation of arbitrary data structures. For more information, see json.org.

## Resources and resource types

A resource is an individual data entity with a unique identifier. The table below describes the different types of resources that you can interact with using the API.